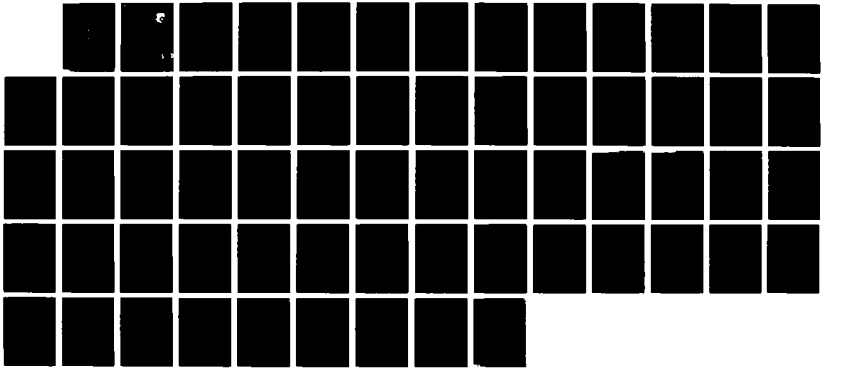
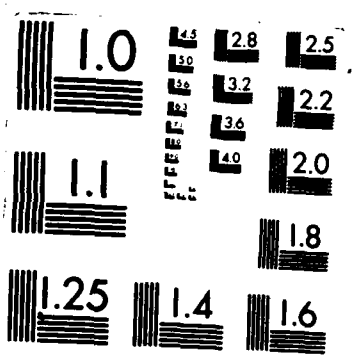
AD-A181 707 INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 5 1/1
COMMON DATA MODEL S..(U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU..
UNCLASSIFIED M L APICELLA ET AL. 01 NOV 85 PS-620141212 F/G 12/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

DTIC FILE COPY

AFWAL-TR-86-4006
Volume V
Part 12

AD-A181 707

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 12 - NDML Precompiler Parse Procedure Division Product Specification

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985

November 1985

Approved for public release; distribution is unlimited.

DTIC
ELECTE
JUN 30 1987
S D
A

PREPARED FOR:

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

87 6 30 000

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

1 November 1985

REPORT DOCUMENTATION PAGE

AD-A181 767

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-86-4006 Vol V, Part 12
6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFWAL/MLTC
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345	7b. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF 8b. OFFICE SYMBOL (If applicable) AFWAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5155
8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433	10 SOURCE OF FUNDING NOS PROGRAM ELEMENT NO. 78011F; PROJECT NO. 7500; TASK NO. 62; WORK UNIT NO. 01
11. TITLE (Include Security Classification) (See Reverse)	
12. PERSONAL AUTHOR(S) Apicella, M. L. and Thompson, D. B.	
13a. TYPE OF REPORT Final Technical Report	13b. TIME COVERED 22 Sept 1980 - 31 July 1985
14. DATE OF REPORT (Yr., Mo., Day) 1985 November	15. PAGE COUNT 62
16. SUPPLEMENTARY NOTATION ICAM Project Priority 6201	The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.

17 COSATI CODES		
FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This document is the product specification establishing the design implementation of the IISS Configuration Item PRE2 which locates and isolates Neutral Data Manipulation Language (NDML) statements and performs necessary semantic checking of the command.

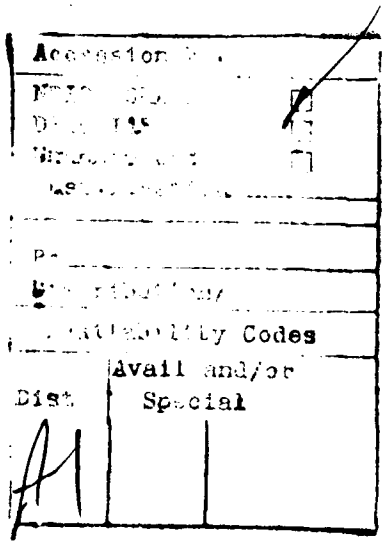
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b TELEPHONE NUMBER (Include Area Code) 513-255-6076 22c OFFICE SYMBOL AFWAL/MLTC

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

11. Title

Integrated Information Support System (IISS)
Vol V - Common Data Model Subsystem
Part 12 - NDML Precompiler Parse Procedure Division
Product Specification



PREFACE

This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Reviewer
D. Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search
General Dynamics/ Ft. Worth	Responsible for factory view function and information models

Subcontractors	Role
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business
North American Rockwell	Reviewer
Northrop Corporation	Responsible for factory view function and information models
Pritsker and Associates	Responsible for IDEF2 support
SofTech	Responsible for IDEF0 support

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

Subcontractors	Role
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

Contractors	ICAM Project	Contributing Activities
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC)

Contractors	ICAM Project	Contributing Activities
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP)
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI)
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

		Page
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-1
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-2
3.3.1	Inputs/Outputs	3-2
3.4	Program Interrupts	3-3
3.5	Timing and Sequencing Description	3-3
3.6	Special Control Features	3-3
3.7	Storage Allocation	3-3
3.7.1	Database Definition	3-3
3.7.1.1	File Description	3-3
3.7.1.2	Table Description	3-3
3.7.1.3	Item Description	3-3
3.8	Object Code Creation	3-3
3.9	Adaptation Data	3-3
3.10	Detail Design Description	3-4
3.10.1	Main Program List	3-4
3.10.2	Module List	3-6
3.10.3	External Routines List	3-8
3.10.4	Include File List	3-10
3.10.5	Where Include File Used List	3-12
3.10.6	Where External Routine Used List	3-16
3.10.7	Main Program Parts List	3-21
3.10.8	Module Documentation	3-23
3.10.9	Include File Description	3-29
3.10.10	Hierarchy Chart	3-41
3.11	Program Listings Comments	3-49
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

SECTION 1

SCOPE

1.1 Identification

This specification establishes the design of Function PRE2, "Parse Procedure Division", one of the major functions of the Configuration Item "Precompiler" to be built and formally accepted by the ICAM Program office. This CI constitutes one of the subsystems of the Common Data Model Processor (CDMP)

1.2 Functional Summary

The purpose of this Computer Program Configuration Item (CPCI) is to identify all Neutral Data Manipulation Language (NDML) commands contained within the Application Program and for each command, insure all other Precompiler activities are performed.

The following functions will be performed by this CPCI:

1. Scan input file for the start of an NDML command.
2. Call the NDML Parser to syntactically check the command.
3. For each command:
    a. Populate the External Schema (ES) action and qualify lists.
    b. Control all other precompiler activities.

SECTION 2

DOCUMENTS

2.1 Reference Documents

1. ICAM Documentation Standards: IDS15012000A, 28 December 1981.

2. D. Appleton Co., CDM Administrator's Manual; UM620141000, March 1984.

3. D. Appleton Co., CDM1-IDEF Model of the Common Data Model; CCS620141000, 15 May 1985.

4. D. Appleton Co., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDML Precompiler; DS620141200, October 1984.

5. D. Appleton Co., Embedded NDML Programmer's Reference Manual; PRM620141200, March 1985.

6. Softech, Inc., NTM Programmer's Guide; UM620140001, July 1984.

7. Control Data Corp., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDDL Command Processor; DS620141100, June 1985.

2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

Common Data Model: (CDM) Describes common data application process formats, form definitions, etc, of the IISS and includes conceptual schema, external, internal schemas, and schema transformation operators.

Data Field: (DF) An element of data in the external schema. It is by this name that an NDML programmer references data.

Database Management System: (DBMS)

Distributed Request Supervisor: (DRS) This IISS CDM subsystem configuration item controls the execution of distributed NDML queries and non distributed updates.

Domain: A logical definition of legal attribute class values.

Domain Constraint: Predicate that applies to a single domain.

External Schema: (ES)

Forms: Structured views which may be imposed on windows or other forms. A form is composed of fields where each field is a form, item, or window.

Forms Processor: (FP) A set of callable execution time routines available to an application program for form processing.

Internal Schema: (IS)

Integrated Information Support System: (IISS) A test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers interconnected via a local Area Network.

Mapping: The correspondence of independent objects in two schemas: ES to CS or CS to IS.

Network Transaction Manager: (NTM) Performs the coordination, communication and housekeeping functions required to integrate the application processes and system services resident on the various hosts into a cohesive system.

Neutral Data Manipulation Language: (NDML) A language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieved and single node updates.

ORACLE: Relational DBMS based on the SQL (Structured Query Language, a product of ORACLE Corp, Menlo Park, CA). The CDM is an ORACLE database.

Parcel: A sequential file containing sections source code of the input application program.

Request Processor: (RP) A COBOL program that will satisfy a retrieval or update NDML subtransaction against a particular Database Management System.

User Interface: (UI) Controls the user's terminal and interfaces with the rest of the system.

Virtual Terminal Interface: (VTI) Performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by UI software which constitutes the Virtual Terminal Definition. Specific terminals are then mapped against the Virtual Terminal software by specific software modules written for each type of real terminal supported.

SECTION 3

REQUIREMENTS

3.1 Structural Description

A graphic portrayal of this CPCI is included in Section 3.10. This chart shows the hierarchical relationships of each module making up this CPCI.

This CPCI uses a number of lower level modules to handle specific operations such as:

1. Open associated files and parcels (OPNERR, OPNFILE, OPNINPT, OPNPAR1, OPNPAR2, OPNPAR3, OPNPAR4).

2. Read associated files and parcels (REDINPT, REDLINE, REDPAR1, REDPAR2, REPAR3, REDPAR4).

3. Write records to associated files and parcels (WRITERR, WRTLINE, WRTPAR1, WRTPAR2, WRTPAR3, WRTPAR4).

4. Close associated files and parcels (CLSERR CLSFILE, CLSINPT, CLSPAR1, CLSPAR2, CLSPAR3, CLSPAR4).

5. Syntactically check the NDML command (NDMLPAR).

6. Semantically check the NDML command and populate external schema action and qualify lists (CDPRE2A).

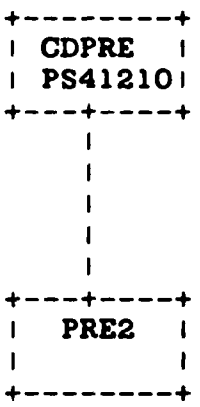
3.2 Functional Flow

This CPCI implements the logic defined in the Development Specification for this CPCI. Details of inputs/outputs and relationships between modules are to be found in Section 3.10.

This CPCI has been designated to operate in a batch or interactive mode. It must operate in the system environment established for IISS; that is, use of the Network Transaction Manager. It must use the ORACLE DBMS installed on a DEC VAX computer.

3.3 Interfaces

The following diagram depicts the interface of PRE2 with other CPCI's on the system.



3.3.1 Inputs/Outputs

The following table depicts the inputs and outputs of this CPCI. A detail description for each item can be found in the DS for this CPCI.

Function: PRE2

INPUT	OUTPUT
Parcel 1 File Name	Code generator table
Parcel 2 File Name	File Status
Parcel 3 File Name	Module Status
Parcel 4 File Name	
ORACLE Logon Data Area	
Error File Name	
Source Language	
Current Host	

Target Host
I/O Section Indicator

3.4 Program Interface

Not applicable to this CPCI.

3.5 Timing and Sequencing Description

This CPCI is called upon by the CDPRE precompiler control module to parse and precompile each NDML statement found in a COBOL procedure division or FORTRAN executable code section. PRE2 remains in control until the end of the user software module is detected. Function PRE1 has previously scanned the input until the beginning of executable host language statements.

3.6 Special Control Features

Not applicable to this CPCI.

3.7 Storage Allocation

3.7.1 Database Definition

The database used by this CPCI is the Common Data Model (CDM) database. This model is defined by the CDM1, the IDEF-1 model of the CDM, Reference Document Number 3. The database was constructed using ORACLE.

3.7.1.1 File Description

No permanent files have been defined for this CPCI. It uses temporary scratch files for program source code.

3.7.1.2 Table Description

All tables used by this CPCI have been defined by the Development Specification for this CPCI.

3.7.1.3 Item Description

Not applicable to this CPCI.

3.8 Object Code Creation

The object code for this CPCI will be created by the system

integration test team by using defined IISS Software Configuration Management procedures. This CPCI will use the COBOL, "C" and FORTRAN language compilers. In addition, source code generated by the YACC and LEX UNIX tools must be compiled to become objects in the final, executable software package.

3.9 Adaptation Data

This CPCI has been coded using ANSI COBOL, FORTRAN and a "standard" subset of the "C" language. The intent was to provide a transportable system. Any system environment supporting these languages, a virtual memory management scheme, the COMM and NTM subsystems of IISS and the ORACLE Database Management System should be able to support this CPCI. Every possible attempt has been made to localize and identify any machine or environment dependent modules through the original design of the IISS and application of Configuration Management Procedures.

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

PARSE PROCEDURE DIVISION Main Program List

Module Name	Purpose
CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

PARSE PROCEDURE DIVISION Module List

Module Name	Purpose
CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

PARSE PROCEDURE DIVISION External Routines List

Module Name	First User
CDMACR	CDPRE2
CDPRE4	CDPRE2
CLSERR	CDPRE2
CLSPAR4	CDPRE2
CPFCOR	CDPRE2A
CPFNXT	CDPRE2A
CPFONE	CDPRE2A
CPFVAL	CDPRE2A
ERRPRO	CDPRE2
LOWUPP	CDPRE2A
NDMLPAR	CDPRE2
OPNERR	CDPRE2A
OPNPAR4	CDPRE2
REDINPT	CDPRE2
RPTERR	CDPRE2A
TOKEN	CDPRE2
UNPLINE	CDPRE2
WRITERR	CDPRE2
WRTPAR4	CDPRE2

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "***** PURPOSE NOT FOUND BY STRIPPER *****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

PARSE PROCEDURE DIVISION Include File List

File Name	Purpose
CGTABLE	CODE GENERATING TABLE- TRACKS ALL GENERATED SOFTWARE
CHKCDM	IISS CDMP CHECK STATUS CODES
ERRCDM	IISS ERROR STATUS CODES FOR CDMP MODULES
ERRPRO	PROCESS ERROR INCLUDE FILE
ESAL	EXTERNAL SCHEMA ACTION LIST
ESQUAL	EXTERNAL SCHEMA QUALIFY LIST
MACDAT	WS VARIABLES FOR MACRO COPY UTILITY
NDMLLST	VALID LIST NUMBERS FOR NDML PARSER
ORCLEDA	WS DEFINITION FOR THE ORACLE LOGIN AREA
SBSTLST	WS DEFINITION FOR THE SUBSTITUTION LIST TABLE
UVABBR	USER VIEW ABBREVIATION LIST

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

PARSE PROCEDURE DIVISION Where-include-file-used List

Include File	Module Name	Module Purpose
CGTABLE	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
CHKCDM	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
ERRCDM	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
ERRPRO	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
ESAL	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
ESQUAL	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION

PARSE PROCEDURE DIVISION Where-include-file-used List

Include File	Module Name	Module Purpose
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
MACDAT		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
NDMLLST		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
ORCLEDA		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
SBSTLST		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
UVABBR		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS

PARSE PROCEDURE DIVISION Where-include-file-used List

Include File	Module Name	Module Purpose

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

PARSE PROCEDURE DIVISION Where-external-routine-used List

System Module	Module Name	Module Purpose
CDMACR		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
CDPRE4		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
CLSERR		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
CLSPAR4		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
CPFCOR		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
CPFNXT		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS

PARSE PROCEDURE DIVISION Where-external-routine-used List

System Module	Module Name	Module Purpose
CPFONE		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
CPFVAL		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
ERRPRO		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
LOWUPP		
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
NDMLPAR		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
OPNERR		
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS

PARSE PROCEDURE DIVISION Where-external-routine-used List

System Module	Module Name	Module Purpose
OPNPAR4	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
REDINPT	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
RPTERR	CDPRE2A	CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
TOKEN	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
UNPLINE	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
WRITERR	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
WRTPAR4		

PARSE PROCEDURE DIVISION Where-external-routine-used List

System Module	Module Name	Module Purpose
	CDPRE2	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more that once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external "routine". The Purpose of the Main Program module is listed as well.

PARSE PROCEDURE DIVISION Main Program Parts List

Main Pgm Name	Module Name	Module Type
CDPRE2	Purpose--›	PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
	CDMACR	External routine
	CDPRE2A	Well-defined module
	CDPRE4	External routine
	CLSERR	External routine
	CLSPAR4	External routine
	CPFCOR	External routine
	CPFNXT	External routine
	CPFONE	External routine
	CPFVAL	External routine
	ERRPRO	External routine
	LOWUPP	External routine
	NDMLPAR	External routine
	OPNERR	External routine
	OPNPAR4	External routine
	REDINPT	External routine
	RPTERR	External routine
	TOKEN	External routine
	UNPLINE	External routine
	WRITERR	External routine
	WRTPAR4	External routine

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as otained

from the source code.

ARGUMENTS:	The arguments with which this routine is called if it is a Subroutine or a Function.
INCLUDE FILES:	A list of all the files that are included into this module as well as their purposes.
ROUTINES CALLED:	Subroutines or Functions, either documented or external, called by this module, if any.
CALLED DIRECTLY BY:	The documented routines which call this module, if any.
USED IN MAIN PROGRAM(S):	The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

PARSE PROCEDURE DIVISION Module Documentation

NAME: CDPRE2
PURPOSE: PARSE THE PROCEDURE DIVISION OF USERS APPLICATION
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CDPRE2
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41212

DESCRIPTION:

- CDPRE2 WILL DO THE FOLLOWING:
-

ARGUMENTS:

PARCL1 = DSPLY [X(30)]
PARCL2 = DSPLY [X(30)]
PARCL3 = DSPLY [X(30)]
PARCL4 = DSPLY [X(30)]
ORACLE-LDA = RECRD
ERROR-FILE = DSPLY [X(30)]
SOURCE-LANGUAGE = DSPLY [X(10)]
MY-HOST = DSPLY [X(3)]
TARGET-HOST = DSPLY [XXX]
CODE-GENERATOR-TABLE = RECRD
IOSECTION-INDICATOR = DSPLY [9]
FILE-STATUS = DSPLY [S9(9)]
MODULE-STATUS = DSPLY [X(5)]

INCLUDE FILES:

MACDAT - WS VARIABLES FOR MACRO COPY UTILITY
SBSTLST - WS DEFINITION FOR THE SUBSTITUTION LIST TABLE
ESQUAL - EXTERNAL SCHEMA QUALIFY LIST
ESAL - EXTERNAL SCHEMA ACTION LIST
UVABBR - USER VIEW ABBREVIATION LIST
ERRCDM - IISS ERROR STATUS CODES FOR CDMP MODULES

```
  CHKCDM      - IISS CDMP CHECK STATUS CODES
  ORCLEDA     - WS DEFINITION FOR THE ORACLE LOGIN AREA
  CGTABLE     - CODE GENERATING TABLE- TRACKS ALL GENERATED
                SOFTWARE
  ERRPRO      - PROCESS ERROR INCLUDE FILE
```

ROUTINES CALLED:

```
  OPNERR
  CLSERR
  CLSPAR4
  NDMLPAR
  CDPRE2A     - CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER
                LISTS
  CLSERR
  CLSPAR4
  CDPRE4
  OPNPAR4
  TOKEN
  UNPLINE
  REDINPT
  CDMACR
  WRTPAR4
  WRITERR
  ERRPRO
```

PARSE PROCEDURE DIVISION Module Documentation

NAME: CDPRE2A
PURPOSE: CDPRE2A - PARSE NDML AND BUILD ES PRECOMPILER LISTS
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: CDPRE2A
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41212

DESCRIPTION:

-

THIS MODULE CONTROL THE LOGIC OF PARSING NDML STATEMENTS IN THE USER APPLICATION PROGRAM.

ARGUMENTS:

COMMAND-NO = DSPLY [S9(9)]
NDML-STACK = RECRD
ES-QUALIFY-LIST = RECRD
ES-ACTION-LIST = RECRD
UV-ABBR-LIST = RECRD
ERROR-FILE = DSPLY [X(30)]
RETURN-STATUS = DSPLY [S9(9)]

INCLUDE FILES:

NDMLLST - VALID LIST NUMBERS FOR NDML PARSER
ESQUAL - EXTERNAL SCHEMA QUALIFY LIST
ESAL - EXTERNAL SCHEMA ACTION LIST
UVABBR - USER VIEW ABBREVIATION LIST

ROUTINES CALLED:

CPFVAL
CPFNXT
LOWUPP
CPFONE
CPFCOR
CLSERR
RPTERR
OPNERR

CALLED DIRECTLY BY:

CDPRE2 - PARSE THE PROCEDURE DIVISION OF USERS APPLICATION

USED IN MAIN PROGRAM(S):

CDPRE2 - PARSE THE PROCEDURE DIVISION OF USERS APPLICATION

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: CGTABLE
PURPOSE: CODE GENERATING TABLE- TRACKS ALL GENERATED SOFTWARE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CODE GENERATOR TABLE, HOLDS PERTINENT RESULTS
ABOUT ALL CODE GENERATED OR MODIFIED BY THE
PRECOMPILER
NOTE: ROW 100 RESERVED FOR SWAPPING DURING SORTING.

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: CHKCDM
PURPOSE: IISS CDMP CHECK STATUS CODES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CONTAINS ALL STATUS CODES FOR THE
CDMP MODULES

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: ERRCDM
PURPOSE: IISS ERROR STATUS CODES FOR CDMP MODULES
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CONTAINS ALL ERROR CODES USED BY CDMP *
MODULES FOR ERROR HANDLING *

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: ERRPRO
PURPOSE: PROCESS ERROR INCLUDE FILE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: ESAL
PURPOSE: EXTERNAL SCHEMA ACTION LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CONTAINS THE EXTERNAL SCHEMA INFORMATION FOR AN NDML REQUEST

THE EXTERNAL SCHEMA ACTION LIST

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: ESQUAL
PURPOSE: EXTERNAL SCHEMA QUALIFY LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

CONTAINS EXTERNAL SCHEMA INFORMATION FOR THE NDML QUALIFICATION

THE EXTERNAL SCHEMA QUALIFY LIST

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: MACDAT
PURPOSE: WS VARIABLES FOR MACRO COPY UTILITY
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: NDMLLST
PURPOSE: VALID LIST NUMBERS FOR NDML PARSER
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: ORCLEDA
PURPOSE: WS DEFINITION FOR THE ORACLE LOGIN AREA
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

THE ORACLE LOGON DATA AREA

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: SBSTLST
PURPOSE: WS DEFINITION FOR THE SUBSTITUTION LIST TABLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

SUBSTITUTION-LIST REPRESENTS THE INPUT TABLE
OF SUBSTITUTION PARAMETERS FOR THE CDMACR
MACRO EXPANSION SUBROUTINE

PARSE PROCEDURE DIVISION Include File Description

FILE NAME: UVABBR
PURPOSE: USER VIEW ABBREVIATION LIST
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

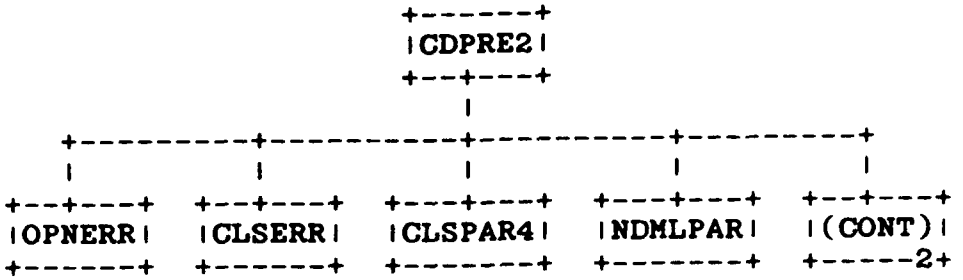
CONTAINS THE ABBREVIATIONS FOR ALL USER
VIEW REFERENCED IN THE NDML REQUEST

3.10.10 Hierarchy Chart

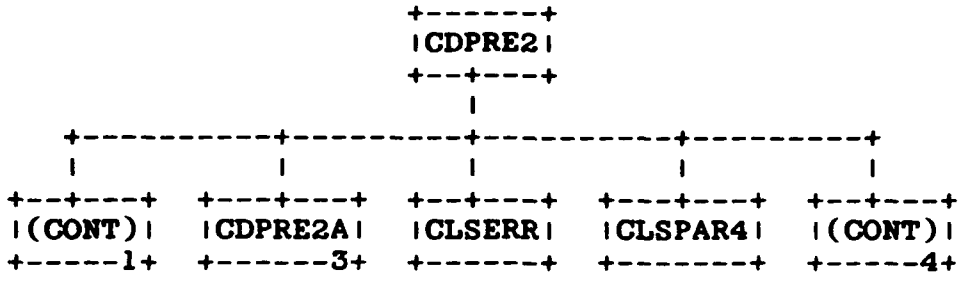
The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

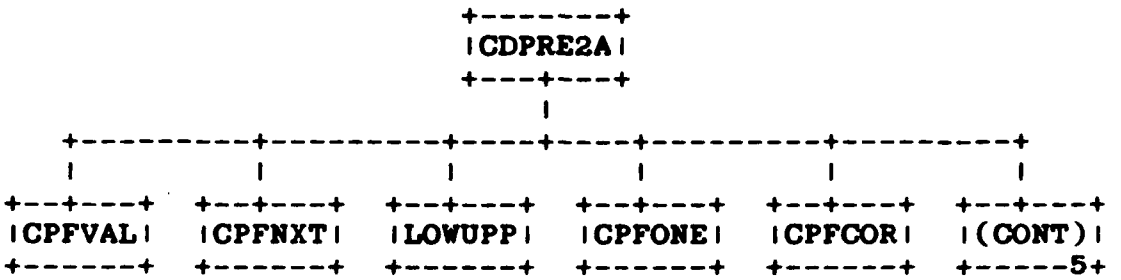
1



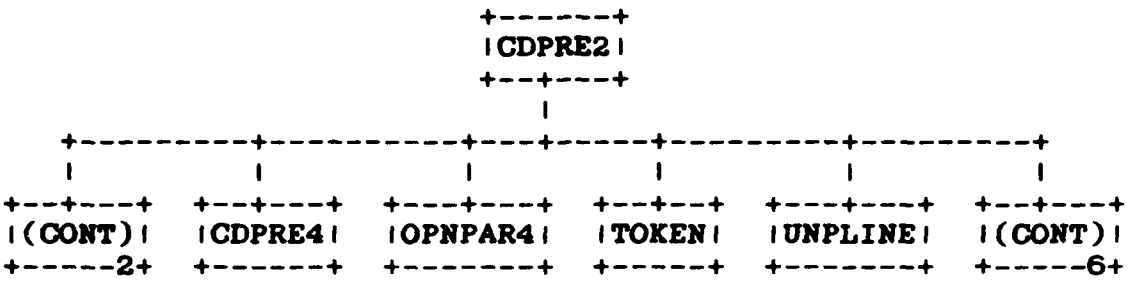
2



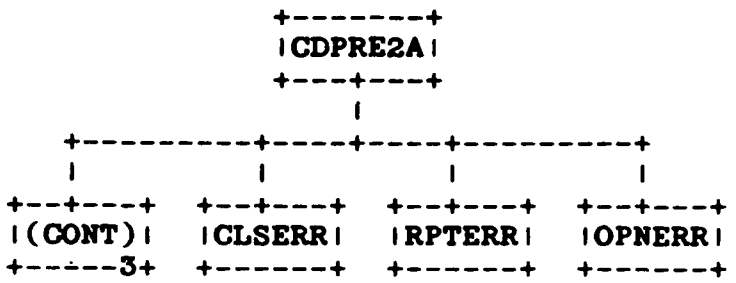
3



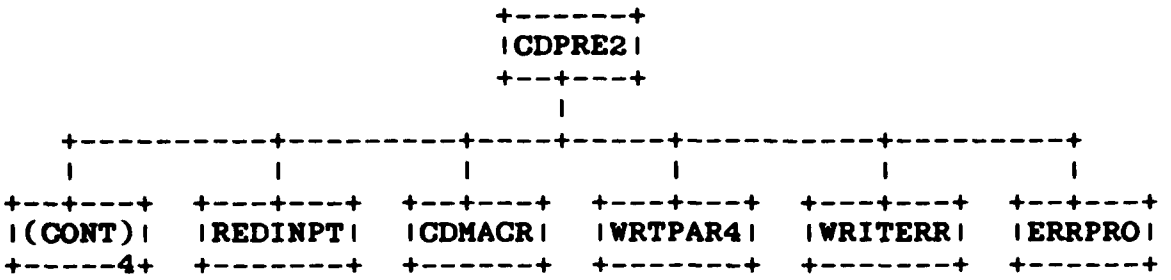
4



5



6



CDMACR
CDPRE2.....1
CDPRE2A....3
CDPRE4
CLSERR
CLSPAR4
CPFCOR
CPFNXT
CPFONE
CPFVAL
ERRPRO
LOWUPP
NDMLPAR
OPNERR
OPNPAR4
REDINPT
RPTERR
TOKEN
UNPLINE
WRITERR
WRTPAR4

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

*U.S.Government Printing Office: 1987 748-061 60899

END

7-87

DTIC